

Integration of domain-specific IT processes and tools in IBM Service Management

N. Joshi
W. Riley
J. Schneider
Y.-S. Tan

In this paper we focus on the integration of domain-specific information technology (IT) processes and tools in the IBM Service Management architecture, a service-oriented software architecture that automates and simplifies the management of IT services. The IT processes are based on a generalized concept of service management that incorporates best practices, such as those defined by the Information Technology Infrastructure Library® (ITIL®). The IT tools are the operational management tools in various domains, such as monitoring, network management, and provisioning. We refer to implementation of IT processes as Process Managers. We first describe three typical scenarios in which integrating the domain-specific IT processes through the use of PMs increases the level of automation. Then, we illustrate the benefits to be gained from integrating IT processes and tools and describe the design of four PMs: the Service Level Process Manager, the IBM Tivoli Availability Process Manager, the IBM Tivoli Capacity Process Manager, and the IT Service Continuity Process Manager.

As information technology (IT) organizations adopt best practices, such as the Information Technology Infrastructure Library** (ITIL**) Service Support and Service Delivery processes, it becomes increasingly important to coordinate information and activities within the organization or across organizational boundaries.^{1,2} Processing what seems to be a request for a simple change to the IT environment can have drastic impact on service-level agreements (SLAs), utilization of existing capacity, and most important, business operations. Integrating domain-specific IT processes such as configuration management and change management becomes critical. For example, a change management process that makes use of resource dependency information

associated with agreements created in the service-level-management (SLM) process and capacity plans defined in a capacity management process is much more effective.

The IBM rendition of the IT service management strategy outlines four types of offerings³:

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/07/\$5.00 © 2007 IBM

1. IBM Tivoli* Unified Process (ITUP)⁴ combines best practices from industry-wide specifications of IT best practices, such as ITIL,^{1,2,5} ISO 20000,⁶ eTOM** (Enhanced Telecommunications Operations Map**),⁷ Six Sigma**,⁸ and COBIT** (Control Objectives for Information and Related Technology),⁹ and proposes a process framework that incorporates the best from each.
2. An operational management product (OMP) is a system that can be used to manage a particular domain or aspect of the IT infrastructure. An OMP provides capabilities such as monitoring, provisioning, automation, and diagnosis for a particular domain, for example, computer systems, networks, software applications, transaction systems, and business services. OMPs for systems and network management are exemplified by IBM Tivoli Monitoring, IBM Tivoli NetView*, and IBM Tivoli Netcool* Omnibus.¹⁰
3. An ITSM (information technology service management) platform is a foundational framework that enables process automation as well as data, functional, and user interface (UI) integration among OMPs and other tools. The IBM Tivoli Change and Configuration Management Database (CCMDB) forms a core part of the ITSM platform.³
4. A Process Manager (PM) is a system for providing and managing the execution of a process workflow by leveraging the services offered by the ITSM platform. It operates the customized workflow, ensuring seamless information flow between the underlying management systems—OMP as well as other process implementations. PMs provide workflow management, work scheduling, task assignment, task tracking, auditing and so on. The IBM Service Management architecture³ relies on service-oriented architecture (SOA)¹¹ and external standards¹² for implementing PMs and enabling integration with OMPs.

IBM is developing or has developed several PMs for ITIL processes such as change management, configuration management, availability management, release management, capacity management, service continuity management, and SLM, with the goal of putting ITIL processes into action.¹³ These PMs exploit the vast array of existing OMPs and the available SOA middleware capability to increase the level of automation and thus make IT organizations more efficient in running their business.

By employing various PMs, customers can apply ITIL and ITUP best practices without starting from scratch. The PMs provide concrete templates for processes (as Web Services Business Process Execution Language [WS-BPEL] files), activities, and value-added tasks for customers to implement their processes. Customers have a starting point rooted in industry best practices and a way to customize the process templates, activities, and tasks to suit their business processes, organizational culture, and boundaries. With PMs, customers can select the appropriate activities and tasks and invoke them from their existing tools. This flexibility, coupled with the widely accepted SOA approach, makes it easier for customers to adopt this approach and elevate their IT operations to higher degrees of automation.

In this paper, we examine cross-domain process-integration scenarios and the integration techniques used to implement them. We also introduce four PMs closely related to ITIL Service Delivery processes: IBM Tivoli Service Level Process Manager, IBM Tivoli Availability Process Manager, IBM Tivoli Capacity Process Manager, and IBM Tivoli IT Service Continuity Process Manager. These PMs are selected for the business importance of the corresponding ITIL processes and the relative novelty of the automation solutions available for these processes.

The rest of this paper is organized as follows. The next section contains three scenarios in which integration of domain-specific IT processes using PMs would increase the level of automation. In the following section we describe SOA-based techniques for integration of PMs and discuss the challenges faced in achieving integration and process automation. The next four sections deal with the implementation of the aforementioned four PMs. Each section introduces the processes, highlights best practices implemented as part of the PM, and describes the benefits of the approach. In the last section we summarize the main ideas.

PM INTEGRATION SCENARIOS

Each PM described in this paper provides activities and tasks specifically focused on offering guidance and assistance when the processes supported by the PM are performed. The PMs have been designed and built to provide data and process integration with other PMs. This allows the data collected in one

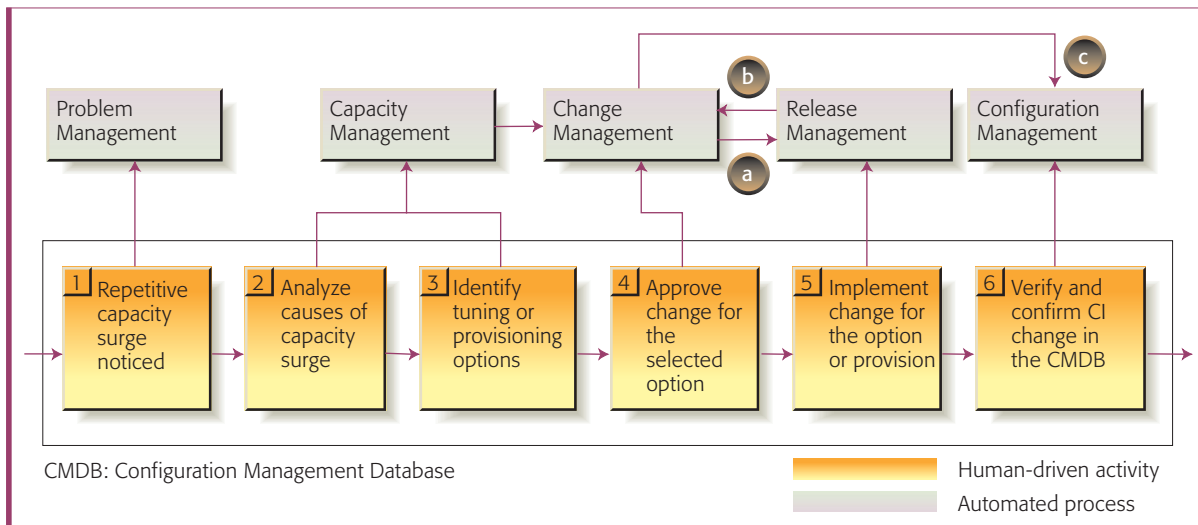


Figure 1
Identifying and resolving a resource capacity problem

process to be leveraged in other processes (e.g., problem management¹), providing increasing benefits as more PMs are used. The PMs also leverage activities across PM boundaries, allowing the processes to be linked across organizational boundaries.

Service-oriented implementations of PMs (described in the next section) involve coordination either in the form of sequential flows or in an interactive fashion. To illustrate the integration of PMs and the resulting benefits, let us examine the following three example scenarios:

1. Identifying and resolving a resource capacity problem, which involves problem management, capacity management, change management, release management, and configuration management and illustrates the operational aspect of systems management.
2. Planning and allocating a resource for a new application, which involves capacity management, SLM, and service continuity management and illustrates the quality-of-service (QoS) aspect of systems management.
3. A service-level feasibility analysis, which involves SLM, change management, availability management, capacity management, and release management and illustrates the process that ensures the feasibility of attaining a specified level of service.

Identifying and resolving a resource capacity problem

Figure 1 illustrates the process flow for identifying and resolving a capacity problem. The sequential flow from steps 1 through 6 illustrates the key activities and related work handled by the IT staff. At step 1, a problem determination specialist notices that an unexpected surge in workload causes an insufficient capacity condition, which has been re-occurring for the past month. The specialist opens a problem ticket, which is routed to the Problem Process Manager for analysis. The Problem Process Manager evaluates the problem and generates a capacity management request at step 2 for a capacity analyst to study. The IBM Tivoli Capacity Process Manager guides the analyst and helps identify a number of tuning and provisioning options at step 3. The analyst discusses the options through a review protocol, and a selection is made. The Capacity Process Manager then automatically generates a request for change (RFC), which is routed to the Change Process Manager.

At step 4, the Change Process Manager triggers the approval of the RFC request and passes it to the IBM Tivoli Release Process Manager to implement the change at point *a*. The Release Process Manager guides the release team through the implementation of the change at step 5, which calls for the provisioning of new server resources in this example. When the release implementation is complete,

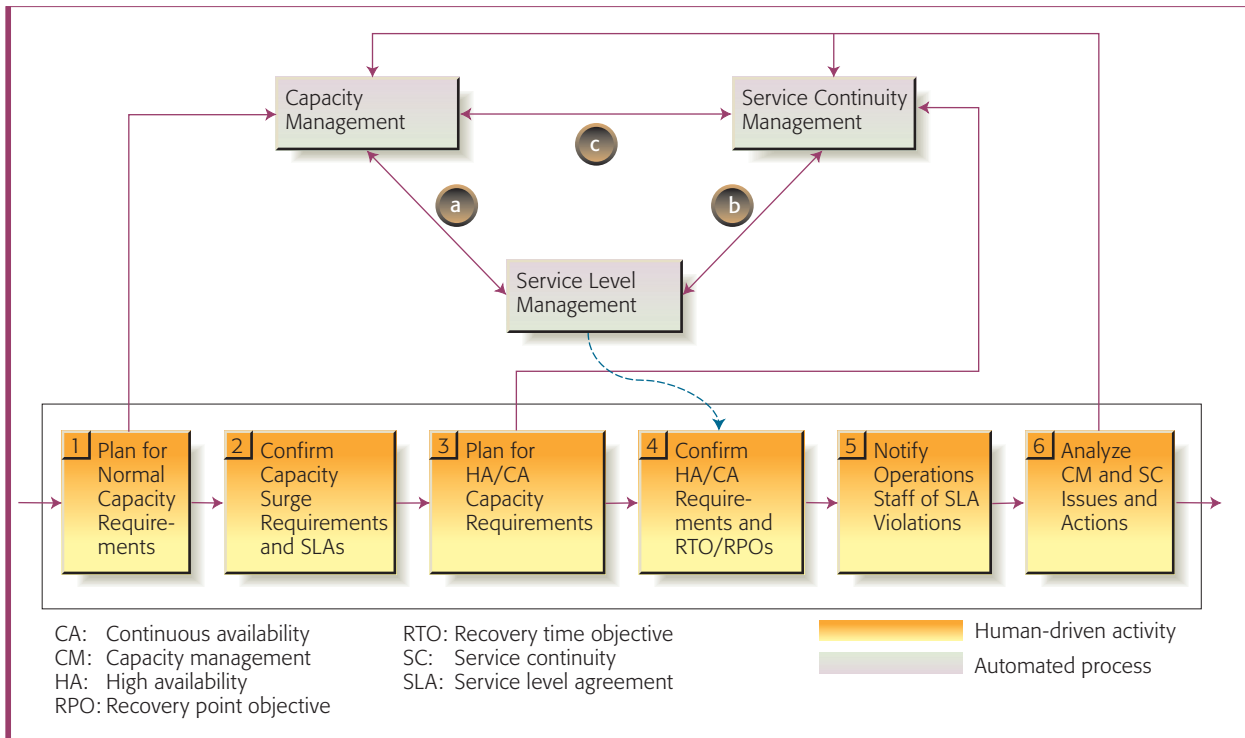


Figure 2
Planning and allocating a resource for a new application

the Release Process Manager notifies the Change Process Manager at point *b* to invoke the IBM Tivoli Configuration Process Manager to update the CMDB (configuration management database) for the affected configuration item (CI) information at point *c*.

Finally, the configuration management staff verifies and confirms the update of CMDB at step 6. The problem ticket is then closed by a routine sequence kicked off at step 6.

Planning and allocating a resource for a new application

Figure 2 illustrates the process flow for planning and allocating a resource for a new application. The sequential flow from steps 1 through 6 illustrates the key activities performed by the IT staff in order to manage the QoS objectives of the application.

At step 1, an application specialist planning the capacity requirements of the new application submits a request to the Capacity Process Manager. The Capacity Process Manager notifies the Service Level Process Manager (step *a* in Figure 2) of the SLA objectives at point *a*. Conversely, the Capacity Process Manager can also receive predefined SLA

objectives from the Service Level Process Manager. The Capacity Process Manager estimates the capacity requirements and confirms the results with the submitting application specialist at step 2 after other stakeholders have approved the results. The Capacity Process Manager then guides a high-availability (HA) and continuous-availability (CA) specialist for additional capacity requirements at step 3.

Steps 3 and 4 are similar to the normal capacity-requirements planning flow, except that there are different SLA objectives and recovery considerations. The specialist submits the request to the Service Continuity Process Manager to plan for an HA/CA strategy, including capacity requirements in a recovery mode. The Service Continuity Process Manager needs to take into account the recovery time objectives (RTOs) and the recovery point objectives (RPOs), which affect both capacity sizing and the performance-oriented SLA objectives of the application in recovery mode. The Service Continuity Process Manager notifies the Service Level Process Manager of the input RTO/RPO objectives at point *b*. Conversely, the Service Continuity Process

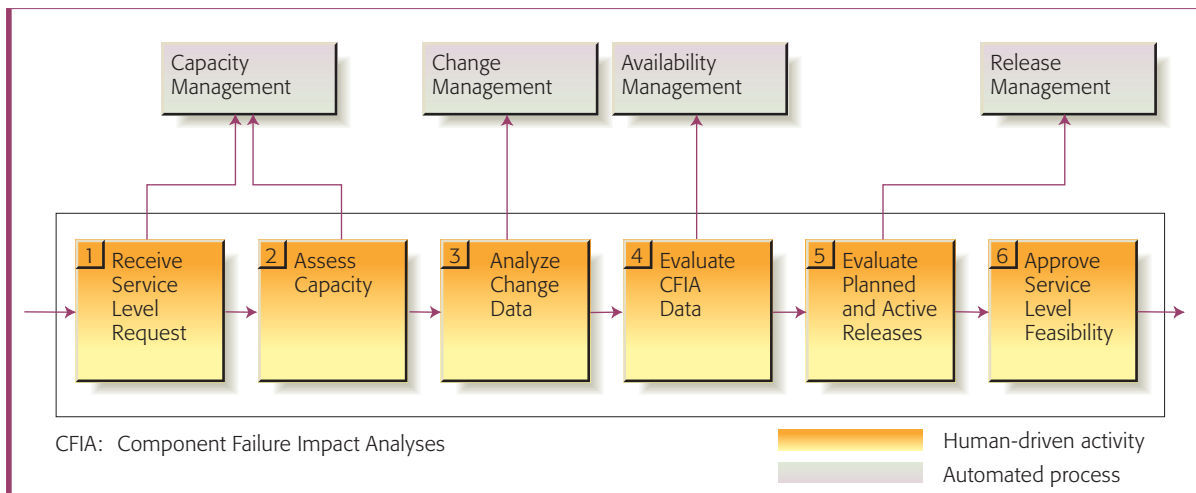


Figure 3
Service-level feasibility analysis

Manager can also receive predefined RTO/RPO objectives from the Service Level Process Manager. The Service Continuity Process Manager then uses the Capacity Process Manager to size the recovery capacity at point c. Conversely, the Service Continuity Process Manager can also receive from and use predefined capacity multipliers from the Capacity Process Manager. After sizing, the Service Continuity Process Manager resynchronizes with the Service Level Process Manager on not only the RTO/RPOs but also the performance-oriented SLAs of the application in recovery mode. It then confirms the results with the submitting HA/CA specialist at step 4 after other stakeholders have approved the results. This ends the planning phase of the example.

Next comes the operational time. Because the Service Level Process Manager handles various types of SLAs, it sets up proper alerts to notify the operations staff of informational, warning, and emergency issues at step 5. These issues are filtered through the normal problem-determination and incident-management process and will be further analyzed at step 6 in the service-continuity-management or capacity-management process. The results of these analyses trigger the appropriate actions at step 6.

Service-level feasibility analysis

Figure 3 illustrates the process flow for a service-level feasibility analysis performed before an SLA

contract is approved. At step 1, a specific service level has been requested by a user of a service. As part of the process of establishing a formal SLA, the feasibility of achieving the requested service level must be determined. The capacity management process is invoked by using the appropriate information from the service level request. In step 2, the service-level feasibility process receives the results of the capacity analysis performed in the capacity management process. The service delivery analyst can end the feasibility process at this step if the current capacity is not sufficient to support the requested service level.

Step 3 invokes specific tasks in the change management process to identify the number, frequency, and downtime associated with changes for the CI needed to support the service level. These are process artifacts, gathered by the change management process, that are used by the service delivery analyst to evaluate any risk associated with committing to the requested service level. In step 4, the CIs needed to support the requested service level are used to invoke the component-failure impact analysis (CFIA) task in the availability management process. The results of the CFIA are used by the service delivery analyst to ensure that there are no critical CI dependencies that have not been properly accounted for in the feasibility-analysis activities.

At step 5, specific tasks in the release management process are invoked to provide information on in-

flight and future planned releases against the CIs needed to support the service level. This information allows the service delivery analyst (or his or her appropriate manager) to understand risks associated with releases that might influence the feasibility of achieving the requested service level. Using the information from the process interactions, the service delivery manager is able to endorse or reject the service level request based upon the feasibility analysis performed. At step 6, the service level feasibility is approved or rejected, moving the service level request to the negotiation SLM activity if rejected or to the activities associated with implementing the service level request if approved.

INTEGRATION TECHNIQUES

This section describes the various service-oriented techniques employed by PMs and OMPs to achieve integration, the benefits of which include (1) effective utilization of OMP data, (2) seamless end-user experience, and (3) automation of human tasks. We briefly describe the important design patterns and techniques used to obtain the needed integration (see Reference 14 for additional details).

System integration module

The system integration module (SIM) is a service-oriented design pattern that enables the functional and data integration of OMPs and PMs. It encapsulates OMP-specific protocol, security, data-access, and mapping aspects for PMs that take advantage of an OMP and provides a well-defined Web Services Description Language (WSDL) interface (known as a logical management operation—LMO) for invocation of needed OMP functions. For example, the IBM Tivoli Availability Process Manager leverages SIMs to obtain operational status through the `getStatus` LMO, from IBM Tivoli Monitoring, IBM Tivoli OMEGAMON*, IBM Tivoli Composite Application Management for Response Time Tracking, and IBM Tivoli Enterprise Console. Multiple LMOs are provided across a wide array of OMPs and used in automation of various process scenarios, allowing the PM to use multiple OMPs with specific OMP knowledge. In effect, the LMOs provide critical pieces of the autonomic-computing MAPE-K (monitoring, analyzing, planning, and executing through knowledge) loop.^{15,16}

Discovery Library Adapter and data federation

Data integration is achieved either through data discovery through the IBM Tivoli Discovery Library

Adapters (DLAs) or through data federation. DLAs support the import and export of resource and relationship data known to OMPs in a standard format—Identity Markup Language (IDML).¹⁷ OMPs export managed data in IDML format, which is then bulk loaded into CMDB and reconciled to form a consistent view of the IT infrastructure. The data loaded by DLAs retains the identification of the OMP that supplied the IDML in order to enable the PMs to direct LMOs to that OMP.

Data federation provides a means to access data from remote sources in a local context. Typically implemented with CMDB, data federation is extensively used to display extended attributes of CIs, where the extended attributes are resident in a remote database. For example, all SLAs stored in Service Level Process Manager are not created locally (many of them are imported). The PM must use data federation to obtain SLA attributes from remote SLA stores, such as IBM Tivoli Service Level Advisor.

Process interface

Processes are composed of activities (subprocesses) and tasks.¹⁴ Activities and tasks that are enabled for integration are required to make a formal specification of their interface accessible. PMs specify the processes using WS-BPEL and implement them in SOA-enabled middleware (such as WebSphere Business Integration¹⁸). The process interfaces are typically implemented as Web Services, which adhere to a WSDL specification for the entry and exit operations. The process interface specifies how to invoke a specific activity or task and whether the interaction with the task is automatic or manual. Once invoked, the activity or task execution is controlled by a process engine (such as WebSphere Process Server¹⁸). Before a system can invoke a process or communicate with it, the system must become a client for the process interface.

Launch and land in context

In switching between a PM and OMPs, a seamless end-user experience is one of the most important integration goals. *Launch in context* allows an end user to launch a target product from the source product and pass it the appropriate context. The passed context allows the target-product user interface to perform the needed operations and navigation to land the user at the proper user-interface function, known as *land in context*. For

PMs, the launch source is the PM and the launch target is an OMP. This capability provides a means to visually take an end user from the PM screen to an OMP screen in the context of a CI. This gives the user the sense that he or she is using a tightly integrated suite of products.

There are additional integration techniques, such as data synchronization, common request management, alerting and notification, and the escalation framework, that are not described here but can assist in cross-domain integration.³

Implementation experience

When implementing the integration of IT processes and tools, we had to overcome the challenge of dealing with a large number of diverse OMPs. These are different products, developed at different times by different vendors. These products needed to be seamlessly integrated despite their different data models, data formats, and interfaces. The use of a service-oriented approach and the previously described integration techniques helped. Another contributing factor was the use of the common data model (CDM) and the database federation in the CMDB.¹⁷

Naming conventions for resources vary widely and are usually incompatible with each other or with industry practices for naming and identification. This problem is further exacerbated by resource managers (OMPs) which employ product-specific conventions for naming and identification. We addressed this challenge by employing CDM, which provides naming rules for resources. CDM can be viewed as an IBM internal standard for data, representing resources and relationships among them; its design is based on industry standards such as Common Information Model (CIM).¹⁹ The IBM CMDB solutions implement CDM and provide the means to import and export CDM-compliant Extensible Markup Language (XML) documents. As long as data sources follow the naming conventions, identification and name-based reconciliation of resources is possible.

When data from a plethora of sources is brought into CCMDDB, the greatest challenge is to reconcile the data so that the same CI is not represented multiple times in the same database. This challenge is addressed by employing two levels of data reconciliation: (1) simple name-based reconciliation, which reconciles CI records based on CDM

naming rules, and (2) rule-based reconciliation, which reconciles CI records based on correlation rules defined in IBM Service Management. Rule-based reconciliation is typically performed after the data has been imported into CCMDDB; whereas, name-based reconciliation can be performed as and when the data is being imported.

The boundaries between PMs and OMPs are the core of integration issues. Whereas traditional products are packaged with emphasis on turnkey solutions, this approach does not work for PMs because they have to be customized for each IT organization. A PM can easily fall into the trap of being tied to a specific OMP. Well-defined process interfaces along with clear identification of the purpose of the OMP and capability can alleviate the problem.

Typically, an IT process requires tasks to be performed by various individuals, who need contextual information (we refer to this as process context) in order to understand and efficiently perform the task. Process context is different from general help and operational guidance information because it provides the context of the overall activity, previously performed tasks, and expected time to completion of a task. One simple technique employed by PMs involves brief visualization of the previously performed tasks, expected completion time, and guidance on performing the current task for each task user interface. In addition, we have explored the possibility of employing an expert advice system that monitors process execution and identifies bottlenecks. The knowledge gained by process execution becomes the basis for future process enhancements and automation.

SERVICE LEVEL PROCESS MANAGER

In this section we describe the Service Level Process Manager. This PM provides the process definition, data integration, and artifact creation necessary to implement the ITIL SLM process as described in ITIL Service Delivery.²

As organizations adopt SLM and begin to formalize the agreements associated with SLM, it is critical that they establish a consistent process which ensures that commitments are made only after proper assessment and evaluation of capabilities has been performed. It is equally important to have consistent processes for handling the agreement attainment results and SLA adjudication based upon

those results. We have identified the following obstacles associated with adopting and implementing the ITIL SLM processes:

1. Clear definitions of recommended activities and tasks and of the artifacts to be considered when implementing SLM are required. The Service Level Process Manager activity and task definitions provide a greater level of process detail than is currently documented in the ITIL SLM process documentation. The ITUP SLM process definitions were used.⁴
2. Data and tools are needed to predict with confidence that the organization can commit to a set of SLAs. CCMDB and OMPs were used to provide necessary data and tools.
3. Support is required for collaboration among the multiple roles involved in establishing an agreement. The IBM Service Management process engine was used to ensure engagement of the proper individuals for task completion, reviews, and approvals in a consistent and repeatable process.

The Service Level Process Manager provides process definitions for three SLM processes defined by ITUP: (1) create and maintain SLAs (this encompasses SLA, operation level agreement [OLA], and underpinning contract (UC)), (2) conduct service review, and (3) formulate service improvement plan. The definition of the create-and-maintain-service-catalog activity is considered to be part of the service catalog offering. Similarly, the monitor-and-report-on-SLA-achievement activity is enabled by several domain-specific OMPs that monitor and manage SLAs.

The Service Level Process Manager leverages the data in the CCMDB to assist in performing those activities. Using resource and service dependency relationships helps ensure that agreements accurately cover the proper set of services and resources. In the following subsections, we document specific examples of how CCMDB data and process artifacts created by other PMs are used in the process descriptions. *Figure 4* illustrates the main SLM processes and the activities contained in each.

Create and maintain SLA process

This process contains several task definitions (best practices) for creation and maintenance of agreements. The process definition guides the user through the needed tasks and activities based upon

the type of agreement being created or maintained during the collection of the required process artifacts. Figure 4 shows the activities that are available for the create-and-maintain-SLA process templates.

The key results of performing this activity are: (1) an agreement (SLA, OLA or UC) that has been reviewed, clearly documented, approved, and enacted with appropriate artifacts collected, and (2) identification of the resources and dependencies required to provide the service for which the agreement is being implemented.

Conduct service review process

This process contains the task definitions and artifact collection for performing a service review for completed agreements. The process activities are shown in Figure 4.

The key results of performing this activity are: (1) the SLA attainment results along with customer feedback, incident information, and CI change information from the CCMDB are used as part of service review, and (2) the automatic creation of process artifacts that record the activity that was performed and the data that was utilized to support the activity. This provides essential audit information should questions arise later about the results of the service review.

SLA adjudication process

The SLA adjudication process is not specifically addressed in the ITIL SLM process, but it is a necessary activity to provide for SLM. SLA adjudication deals with the process of adjusting the data or results associated with SLA attainment for a specific period. These adjustments are allowable and necessary when SLA attainment is not achieved and responsibility for not achieving the SLA does not belong to the service provider. Figure 4 shows the adjudication activities.

The key results of performing this activity are: (1) properly approved and documented adjudication needs and results, and (2) automated guidance to determine if adjudication is necessary for an agreement.

Monitor and report on SLA achievement

Availability metrics for the SLA are analyzed in this activity to determine if the agreement was satisfied for the reporting period. The analysis of individual-

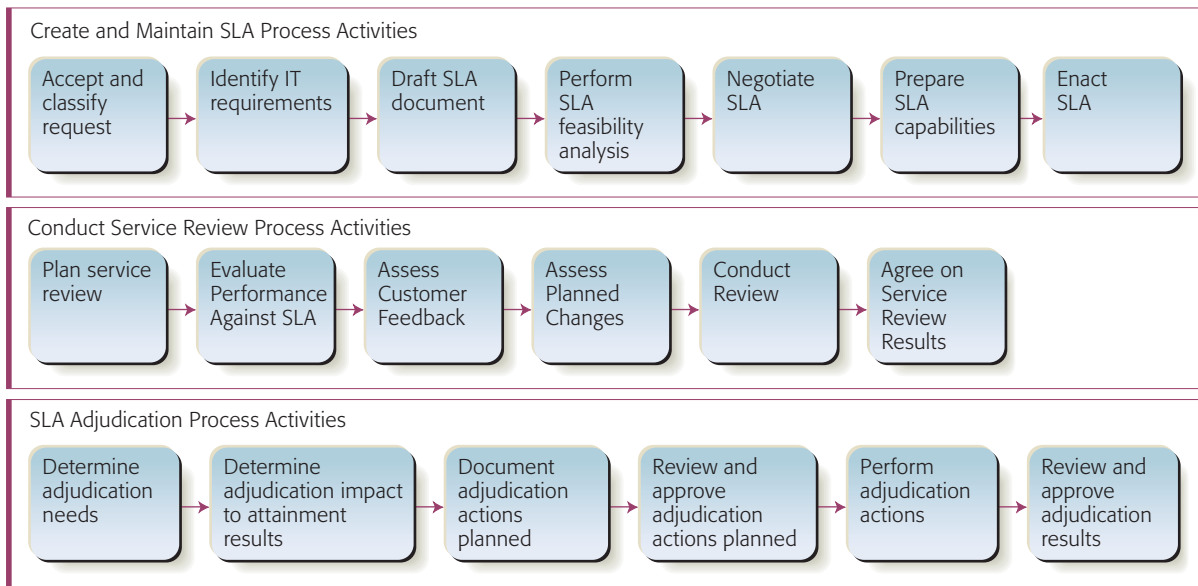


Figure 4
Service Level Process Manager processes and activities

objective attainment is the responsibility of OMPs specializing in SLA monitoring, such as IBM Tivoli Service Level Advisor. The results of this attainment analysis are used as input into the conduct-service-review and formulate-service-improvement-plan processes.

IBM TIVOLI AVAILABILITY PROCESS MANAGER

The availability and performance aspects of the IT infrastructure have a significant impact on the uptime of business services, and thereby on the financial success of the organization. ITIL processes such as incident management, problem management, and availability management address the various aspects of ensuring that the IT infrastructure and the IT services remain operational and in good condition.

As defined by ITIL Service Delivery,² the availability management process is concerned with accurate monitoring of the IT infrastructure, with helping to understand the reasons and impact of service unavailability, and with establishing availability requirements and their translation into implementable improvements to IT infrastructure management. Automation of these aspects has been studied and implemented in a domain-specific manner.²⁰ However automation across domains requires integration across supportive functionality²¹ in the areas

of systems management, network management, application management, and so on. We have identified that automation also requires cross-domain integration with ITIL Service Support¹ processes such as incident management, change management, and problem management, as well as with several IT OMPs.⁹ In addition, we introduce integration with the ITUP-defined event management process.

The IBM Tivoli Availability Process Manager aims at automating critical tasks and activities that span incident management, problem management, and availability management. In the following subsections, tasks and activities provide broad coverage, maximize tool integration, and enable reuse of tasks in various processes: impact analysis, failure analysis, and failure resolution.

These tasks are useful for roles such as IT manager (manages day-to-day operations of the IT organization), availability manager (owns and manages the availability planning, execution, and reporting process), incident manager (manages the incident management process and ensures timely resolution of critical incidents), subject matter expert (SME) (determines what has failed and resolves the failures), and service desk analyst (records and triages the incidents) in the major areas of availabil-

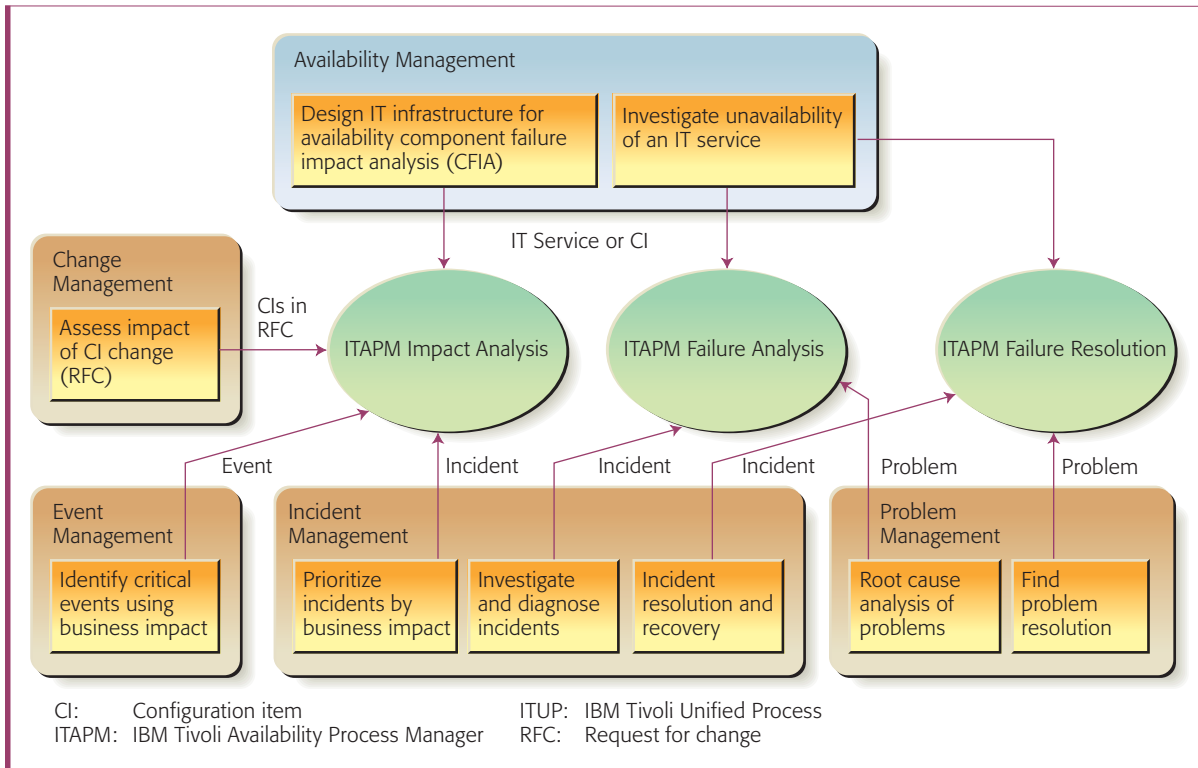


Figure 5
 IBM Tivoli Availability Process Manager tasks in ITUP processes

ity-, incident- and problem-management processes. **Figure 5** depicts how these tasks can be leveraged in the context of various ITUP processes. This figure shows various activities within the processes that can include the impact-analysis, failure-analysis, and failure-resolution tasks. The arrows indicate the data context (event, incident, problem, etc.) in which these tasks are invoked. Next we describe these tasks and their applicability to ITUP processes.

Impact analysis

The impact analysis task focuses on the IT components, component dependencies, and application dependencies that have an existing or potential impact on service-support or service-delivery outcomes. Impact analysis helps automate the component-failure impact analysis (CFIA) activity, which according to ITIL best practices,² can be used in the availability-planning, continuous-availability-improvement, and availability-reporting aspects of the ITIL availability management process. In addition, impact analysis is useful in identifying critical resources.²² In incident management, impact analysis is used to prioritize incidents during the

classification-of-incidents activity. Better prioritization of incidents and identification of the components involved helps route the incident to the appropriate SME and results in faster incident resolution time and better service to customers. In event management, impact analysis service can be used to identify critical events as and when they are reported to the event management tools. This early interception of events saves valuable time in escalating or resolving “failure events.” In the context of change management, when an RFC results in changes to the IT infrastructure, assessing the impact of these changes is an important task before approval of the RFC. The impact analysis service helps the RFC approver assess the impact of the changes.

Failure analysis

The failure analysis task focuses on the investigation of failure, fault isolation, and identification of the root cause of the failure. For availability management, failure analysis is used in the context of an IT service disruption or major CI failure during the investigate-unavailability activity. In incident man-

agement, failure analysis is useful during the investigation and diagnosis of the cause of incidents. Earlier identification of the cause results in faster resolution of incidents. In problem management, historical analysis of events or incidents is performed, in which the main goal is identification of the root cause of the event or incident. Failure analysis helps narrow down the CIs that could be causing the failure and helps determine the root cause.

Failure resolution

The failure resolution task provides ways to resolve the failure. The failure can be reported in the context of an incident or a problem. In incident management, this task is useful during the incident-resolution and recovery activity, whereas in problem management, this task is useful during the control-problems activity. Resolution of failure depends highly on the particular system and situation. Generic automation of all types of failure is highly challenging; however, frequently occurring failures can be documented, and over time, their resolutions can be automated. We use the failure resolution capabilities of domain-specific OMPs by launching the OMP user interfaces in the context of the component and failure situation and by providing contextual help (documentation) to resolve the failure.

IBM TIVOLI CAPACITY PROCESS MANAGER

Efficient capacity utilization impacts the cost and responsiveness of services provided. Whereas capacity planning in the mainframe environment is well understood, the situation is different in a distributed environment, where the significant investment it requires hinders its full adoption.⁴ Thus, capacity management is rare in distributed environments and most businesses either over- or under-plan. Additionally, there is no easy way of sharing excess capacity. This results in high recovery costs and missed opportunities. However, because it is hard to sustain and grow the business if the base capacity is not managed well, various solutions have been proposed.

One cannot expect businesses to adopt standards without substantial assistance. Shrink-wrapped (readymade) vendor solutions could be limiting. Capacity management is knowledge-centric and needs to integrate the environment of the individual business. The Capacity Process Manager implements a “living” Capacity Process Manager model that provides selected ITIL⁵ and ITUP⁴ best practices

as process and activity templates. Task execution details can be provided later after specific knowledge has been harvested. Integration of supporting point products can then follow the tasks. (A point product is an individual product targeted to a particular market segment or domain, for example, ITCAM for WebSphere or ITCAM for J2EE.) The next sections discuss this model in detail.

IBM Tivoli Capacity PM implements the following capacity management best practices from ITUP by means of a set of process templates:

- *Model and size capacity requirements*—determine performance and capacity requirements for an IT solution.
- *Monitor, analyze, and report capacity usage*—measure capacity usage and analyze potential issues to make recommendations and produce reports.
- *Plan and initiate service and resource tuning*—select tuning options and plan for the associated change request.
- *Produce and maintain the capacity plan*—forecast future capacity requirements.

Figure 6 illustrates the core concept of this PM—an iterative refinement loop anchored on a workload profile model. The refinement loop ties individual process templates together for the management of capacity consumed by an application. Tivoli Capacity Process Manager supports a request and response interface and routes a request to a specific process template that runs through a thread of activities; for example, a sizing request for a banking application is routed to the model-and-size-capacity-requirements template. Multiple requests against the same profile affect the same application. As an example, sizing, monitoring, analyzing, and tuning requests can be entered in a natural order, pointing to the same banking application profile. A single iteration can reduce the capacity required from six servers to five.

The profile specifications contain business, service, and resource requirements needed for capacity management: workload definition, performance (service level) objectives, hardware/software prerequisites (machine types, memory, disk, network, operating system, middleware, versions, etc.), architecture topology, deployed capacity (links to related CIs), performance history, and additional

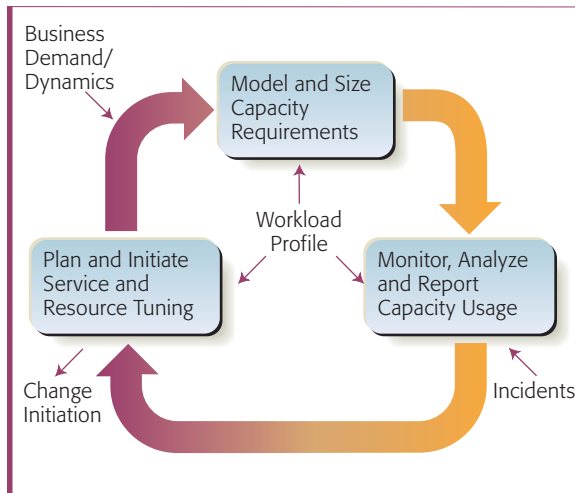


Figure 6
Iterative capacity management process

attributes (e.g., availability, test capacity, backup storage, cost).

The IBM Tivoli Capacity Process Manager uses this information to interact with the CMDB, OMPs, and PMs and to point to solutions for current capacity requirements; for example, a bank teller application owner can initiate the profile with “medium” CPU utilization and “fast” response time as the performance objectives. The monitor-and-analysis process will later help identify the capacity adjustment required to support the workload actually realized, as in the cross-domain integration scenarios previously discussed.

IT SERVICE CONTINUITY PROCESS MANAGER

The business-continuity-management process deals with the likelihood of a disaster, how the disaster interferes with the business process, and how the business can continue to operate. An interruption could be related to a winter storm, the loss of electricity to the general area, or the complete inaccessibility of a facility for an extended period of time. The cause of the interruption is irrelevant; what is important is gaining management control and processing capacity soon after the interruption. The IT Service Continuity Process Manager ensures that the agreed-to IT services continue to support the business in the event of a disruption (disaster) to the business, based on the committed recovery objectives.

The process is required to sustain the vitality of the business, foster close working arrangements between IT and business functions, maintain competitive advantage, and continue to meet regulatory requirements; therefore, the IT Service Continuity Process Manager aims (1) to support the business-continuity-management process and to form the IT part of the business continuity plan of the organization, and (2) to ensure that the predetermined SLAs (recovery objectives) can be met through the recovery of the agreed-upon IT services.

The IT Service Continuity Process Manager fulfills its mission through risk reduction measures, controlled recovery options, and restoration facilities. In general it deals with the additional complexity of a full management cycle for IT redundancy. IT redundancy is required to efficiently “failover” the IT business, and it defines a set of IT resources to be used as backup. This includes not only IT servers, operating systems, middleware, applications, data and storage, but also the hosting locations like buildings and sites. The IT Service Continuity Process Manager encompasses the following:

- Planning of such environments considering costs, business priorities, and risk mitigations.
- Configuration of IT redundancy like HA clusters, storage replication, and data center automation.
- Monitoring and testing of the configured IT redundancy, which includes the simulation of outages. This is particularly important after changes have been applied to the initial configuration.
- Coordination with the incident process (PM or not) in terms of assessing if an incident or set of incidents is a disaster or is to be handled as a normal outage within the incident- and problem-management process.
- Coordination of the various recovery steps throughout the applied configurations, tools, platforms, and people. This includes helping identify the right instant to trigger site failovers (e.g., approval) and the activation of non-IT-related recovery steps (e.g., evacuation of people).

The IT Service Continuity Management PM provides such services according to the following recovery objectives: (1) RTO, which denotes how fast (in time units) the business can recover (in case of site failovers, this time typically includes a term for the

human decision and approval process), and (2) RPO, which denotes how much (committed) data is lost (it reflects the time that the systems have to go back in the data structures—data and data logs—to recover consistently).

Determining the RTO and RPO values requires an understanding of the outage scope (failed and impacted IT resources). For example, customers may decide to support higher-quality RTO and RPO for single IT server outages by using HA cluster solutions; whereas, complete site outages may be outsourced, accepting a recovery time of days rather than hours or minutes.

The IT Service Continuity Process Manager covers the IT aspects in support of business restoration. Often referred to as the disaster recovery plan, it contains the actions to be taken to restore operability of the target system, application, data, or computer facility at an alternate site after an emergency. Furthermore, the IT Service Continuity Process Manager represents a comprehensive statement of actions to be taken before, during, and after a disaster. The planning should be documented and tested to ensure the continuity of operations and availability of critical resources in the event of a disaster.

The probability of a disaster occurring in an organization is highly uncertain. Predictability of the actions taken in response to a disaster is the goal. This can only be accomplished by having a combination of automated functions and well-documented and regularly tested procedures.

The general phases of the service continuity process describe mainly a project refinement cycle, starting from the collection of business requirements, to the IT strategy and the concrete plan for building the IT environment, to the configuration itself, and finally, to the monitoring and execution of the IT recovery in case of a disaster. IBM Service Management activities support the IT-service-continuity management cycle by

- integrating IT platform tools, such as HA clusters and storage replication, and backup/restore facilities as OMPs for the general use of PMs,
- representing IT redundancy as IT topology in the CMDB, including the discovering of existing HA cluster and replication topologies,

- improving the outage simulation and test facilities of some OMPs,
- monitoring the IT environment to signal problems in RTO and RPO configurations (e.g., loss of IT redundancy),
- monitoring the recovery times to indicate whether the applied RTO and RPO has been (can be) achieved, and
- providing best practices workflows as follows: (1) for the detection of a significant outage (versus a normal problem incident), (2) to assist in the configuration of RTO and RPO qualified IT redundancy, and (3) to coordinate recovery steps (IT-related and non-IT-related) in the case of larger site failover scenarios.

Figure 7 depicts the activities and tasks of the IT Service Continuity Process Manager and their interaction with the activities and tasks of other PMs.⁴ As with previously described PMs, the core capability of the IT Service Continuity Process Manager is to facilitate and coordinate the activities that should be followed by the appropriate SMEs and automation tools. For example, as indicated in *Figure 7*, capacity backup plans created in the capacity management process are driven by service continuity requirements.

CONCLUSION

In this paper, we show how PMs translate ITIL best practices into actionable process implementations through ITUP refinement and OMPs. We have described several scenarios in which activities of the various PMs intertwine and opportunities for automation become apparent. We also review a number of integration techniques and demonstrate them in the design of four PMs: the Service Level Process Manager, the IBM Tivoli Availability Process Manager, the IBM Tivoli Capacity Process Manager, and the IT Service Continuity Process Manager.

By utilizing a PM, customers can implement ITIL best practices without starting from scratch. The PMs provide concrete process templates, activities, and tasks for customers to implement their processes, providing a starting point rooted in industry best practices. Customers have the option of customizing the process templates, activities, and tasks to suit their particular business processes and organizational culture. With the PMs, customers can select the appropriate activities and tasks and

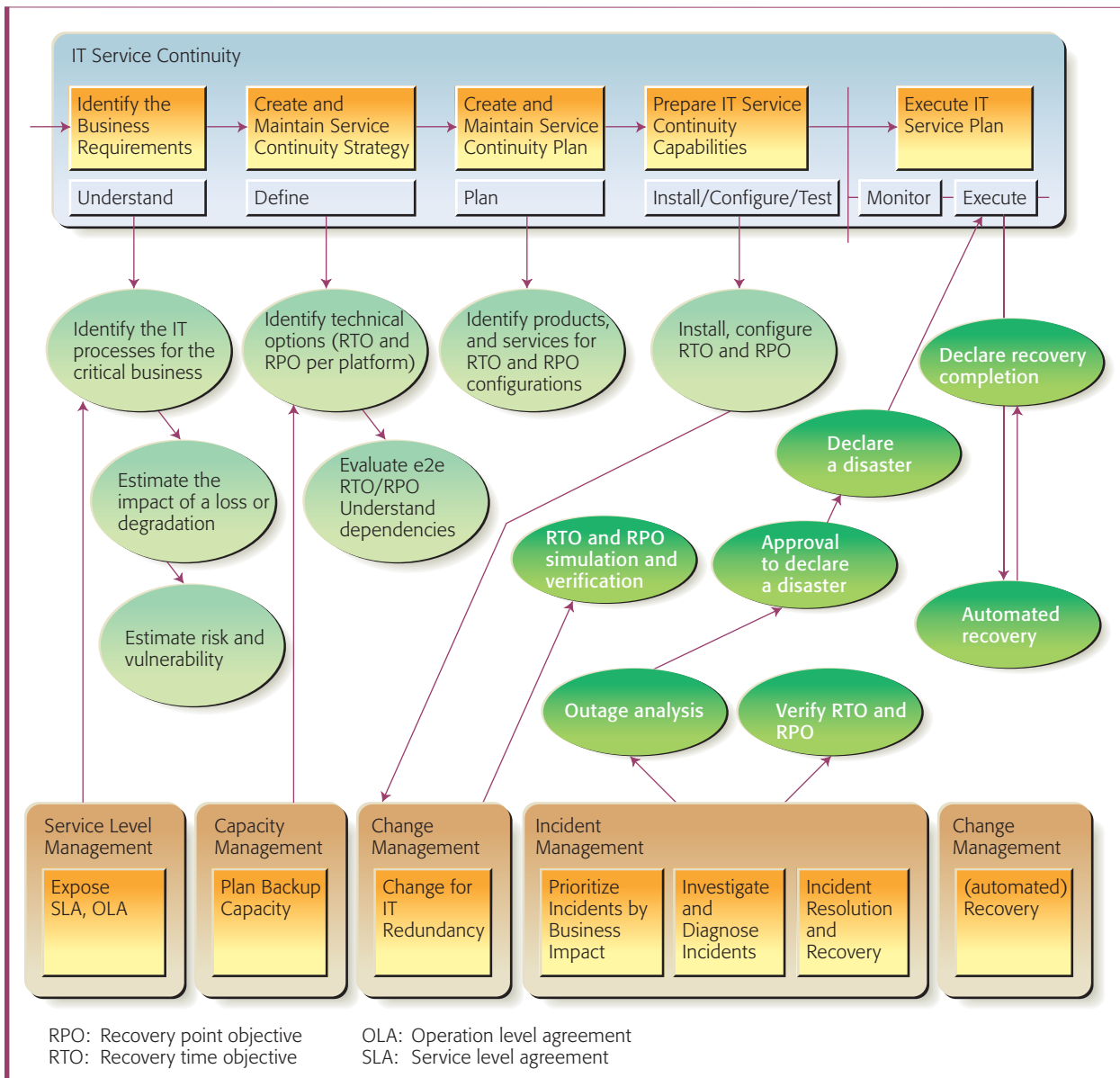


Figure 7
IT Service Continuity Process Manager activities and their interactions

invoke them from their existing tools. This flexibility, coupled with the widely accepted SOA, makes it easier for customers to adopt this paradigm and elevate their IT operations to higher degrees of automation.

To summarize, although specifications of industry best practices (e.g. ITIL) for IBM Service Management provides good common ground for communication, repeatable realization of best practices in heterogeneous IT environments can be enhanced by

PMs by means of cross-domain integration with OMPs. We anticipate that the experience gained from the use of PMs by customers can be used to improve and refine the process templates. Advances in process design and integration technologies should also help streamline the adoption of IBM Service Management.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of, United Kingdom Office of Government Commerce, Telemanagement Forum Corporation, Motorola, Inc., System Audit and Control Association, or Sun Microsystems, Inc., in the United States, other countries, or both.

CITED REFERENCES

1. *Service Support*, IT Infrastructure Library (ITIL), The Stationary Office, Office of Government Commerce, Norwich, UK (2000), <http://www.itil.org.uk>.
2. *Service Delivery*, IT Infrastructure Library (ITIL), The Stationary Office, Office of Government Commerce, Norwich, UK (2001), <http://www.itil.org.uk>.
3. D. Lindquist, H. Madduri, C. J. Paul, and B. Rajaraman, "IBM Service Management Architecture," *IBM Systems Journal* **46**, No. 3, 423–440 (2007, this issue).
4. IBM Tivoli Unified Process, IBM Corporation, <http://www-306.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html>.
5. *ICT Infrastructure Management*, IT Infrastructure Library (ITIL), The Stationary Office, Office of Government Commerce, Norwich, UK (2003).
6. ISO/IEC 20000: Information Technology—Service Management, International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), 2005.
7. Enhanced Telecom Operations Map (eTOM), TeleManagement Forum, <http://www.tmforum.org/browse.aspx>.
8. Fred R. McFadden, "Six Sigma Quality Programs," *Quality Progress* **26**, No. 6, 37–42 (June 1993).
9. M. Ernest and J. M. Nisavic, "Adding Value to the IT Organization with the Component Business Model," *IBM Systems Journal* **46**, No. 3, 387–403 (2007, this issue).
10. IBM Tivoli Information Center, IBM Corporation, <http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>.
11. IBM Systems Journal issue on Service Oriented Architecture (SOA), **44**, No. 4 (2005).
12. M. W. Johnson, A. Hatley, B. A. Miller, and R. Orr, "Evolving Standards for IT Service Management," *IBM Systems Journal* **46**, No. 3, 583–597 (2007, this issue).
13. IBM IT Service Management, IBM Corporation, <http://www-306.ibm.com/software/tivoli/solutions/it-service-management/>.
14. C. J. Paul, "The Process of Building a Process Manager: Architecture and Design Patterns," *IBM Systems Journal* **46**, No. 3, 479–495 (2007, this issue).
15. J. O. Kephart and D. M. Chess, The Vision of Autonomic Computing, http://www.research.ibm.com/autonomic/research/papers/AC_Vision_Computer_Jan_2003.pdf.
16. A. G. Ganek and T. A. Corbi, "The Dawning of the Autonomic Computing Era," *IBM Systems Journal* **42**, No. 1, pp. 5–18 (2003).
17. H. Madduri, et al., "A Configuration Management Database Architecture in Support of IBM Service Management," *IBM Systems Journal* **46**, No. 3, 441–457 (2007, this issue).
18. WebSphere Business Integration Software, IBM Corporation, <http://www.ibm.com/developerworks/websphere/zones/businessintegration/newto/>.
19. Common Information Model (CIM) Standards, Distributed Management Task Force, Inc., <http://www.dmtf.org/standards/cim/>.
20. R. Bhagwan, K. Tati, Y. C. Cheng, S. Savage, and G. M. Voelker, "Total Recall: System Support for Automated Availability Management," *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation*, 2004, pp. 337–350.
21. F. Cristian and S. Mishra, "Automatic Service Availability Management in Asynchronous Distributed Systems," *IEEE Proceedings of 2nd International Workshop on Configurable Distributed Systems*, 1994, pp. 58–68.
22. M. Sailer, A. Hanemann, and D. Schmitz, "A Framework for Failure Impact Analysis and Recovery with Respect to Service Level Agreements," *2005 IEEE International Conference on Service Computing*, pp. 49–58.

Accepted for publication April 2, 2007.

Published online June 27, 2007.

Niraj Joshi

IBM Software Group, Tivoli, 3901 S. Miami Blvd., Research Triangle Park, NC 27709 (njoshi@us.ibm.com). A senior software engineer in the Availability/Business Services Management area of the Tivoli organization and one of the architects of IBM Service Management offerings, Niraj Joshi joined IBM in 1998 and has held several technical leadership positions. His current interests include emerging technologies, process automation, systems management, and user experience. He is a member of the IBM Autonomic Computing and Tivoli Architecture Board and also a member of the IEEE and ACM.

Wayne Riley

IBM Software Group, Tivoli, 3901 S. Miami Blvd., Research Triangle Park, NC 27709 (wriley@us.ibm.com). A senior software engineer in the Tivoli organization, Wayne Riley is a member of the architecture team for the Availability and Business Service Management family of products. He received a B.S. degree in management from Fairleigh Dickinson University in 1987. He joined IBM in 1991 and has held a number of technical leadership roles.

Juergen Schneider

IBM Software Group, Tivoli, Schoenaicher Str. 220, Boeblingen 71032, Germany (schneiju@de.ibm.com). A Senior Technical Staff Member (STSM) in the Tivoli organization, Juergen Schneider joined IBM in 1986 and held several development, design and architecture positions in IBM STG and Tivoli. His expertise is in enterprise systems management with particular focus on automation and high availability. Currently, he is the lead architect for the management and automation of highly available IT services. In this role he is currently managing the implementation of the IBM Tivoli IT Service Continuity Process Manager.

Yih-Shin Tan

IBM Software Group, Tivoli, 3901 S. Miami Blvd., Research Triangle Park, NC 27709 (ystan@us.ibm.com). A Senior Technical Staff Member, Yih-Shin Tan is the lead architect for the IBM Tivoli Capacity Process Manager in the Tivoli Architecture Group. His current interest is technology innovation work on system management. He is an IBM Master Inventor and a member of the IBM Autonomic Computing and Tivoli Architecture Board. ■